



Welcome to

Faculty of Information Technology and Communication Sciences (ITC),
Computing Sciences Unit,
Hervanta campus



Welcome to

Faculty of Information Technology and Communication Sciences (ITC),
Computing Sciences Unit,
Hervanta campus

remember: too much work in companies ruins studies.

We have survived Spring and Summer, or how ?



COMP.SE.100 EN (Autumn 2020) COMP.SE.100 FI (Spring 2021)

Two course implementations; English and Finnish.

Same contents.

Almost same material... still many English slides in the Finnish course.
Not all slides are translated (Finnish news to English, English pictures/screenshots to Finnish).

Kaksi eri toteutusta, englanniksi ja suomeksi.

Sama sisältö.

Melkein sama materiaali... kaikkea en ole vielä ehtinyt käänämään sille toiselle kielelle. Eikä kaikkea voi käänää (esim. suomenkielisiä lehtileikkeitä englanniksi, tai englanninkielisiä tekstititettyjä kuvia/kuvakaappauksia suomeksi).

week	lectures	exam	weekly exercises	project assignment (exercise work)
35	L1: course basics		--- sign to WE groups ---	sign for project = grouping...
36	Project Assignment explained		WE1: intro to requirements	grouping, groups to Moodle
37	L2: Sw Eng in general		WE2: Trello and agile way	group's Trello board ready with product backlog
38	L3: requirements		WE3: feasibility study and stakeholder analysis	working...
39	L4: basic UML diagrams		WE4: requirements	working...
40	L5: more UML diagrams	EXAM-1	WES: UML diagrams - Use case	working...
41	L6: different sw systems	EXAM-2	WE6: UML diagrams - concept/entity and navigation	deadline for 1st phase documentation and presentation
42	examination week		examination week	examination week
43	L7: life cycle models		groups' 1st presentations	groups' 1st phase presentations
44	L8: quality and testing	EXAM-2	WE7: development processes	feedback group-to-group at PRP, from 1st phase
45	L9: project work	EXAM-2	WE8: testing and error reporting	deadline for diagrams first versions
46	L10: project management		WE9: effort estimation	feedback to groups from diagrams (from assistants)
47	L11: open source, APIs, IPR		groups' 2nd presentations	working...
48	L12: embedded systems, IoT	EXAM-3	WE10: delivery contracts and terms of use	working...
49	L13: recap, summary	EXAM-3	---	final delivery of project documentation
50	examination week		examination week	feedback inside group, student-to-student at PRP
51	examination week		examination week	end of game / game over.
	Lectures: Wed at 1415-16.		Weekly exercises:	
			Mon 0815-10	AUTUMN 2020 (1-2. periods)
			Mon 1215-14	are remote/distant learning.
			Tue 0815-10	
			Tue 1415-16	
			Wed 0815-10.	

COMP.SE.100-EN Intro to Sw Eng

Course structure quickly in nutshell

- **lectures** (volunteer), remote at Zoom
 - pdf slides will be at Moodle (at lecture start time)
 - lecture recordings will be available at Panopto
 - **project assignment** (= exercise work), groups of 4 students
 - sign in groups at Moodle
 - **weekly exercises** (volunteer), Zoom and break room groups
 - sign in groups at Moodle
 - no homework, tasks done in small groups at WE
 - **exam**, three small EXAM exams.
-
- 2nd and 3rd examinations (2021) will be "one big" EXAM.

COMP.SE.100-EN (ItSE) Introduction to Software Engineering

Tensu: remember to start Zoom lecture recording, at 1415

Prefer course Moodle over SISU information.

COMP.SE.100 -EN "ItSE"
Introduction to Software Engineering
2020, 1-2. periods

5 credit units

Notice: trust Moodle over SISU

Well... sorry to inform you, that **SISU may give a bit misleading information**, or may not work as you think. Teachers may not have known how to write course information in the many fields in Spring (Feb..March).

Last year we had 50 signees, 47 students at Moodle, about 29 who actually took this course. We planned Autumn 2020 by that way (with slight raise of attendee numbers).

On 20.08. we had 173 signees at SISU, 92 students at Moodle.
On 25.08.2020; 186 students at Moodle.

Please trust course Moodle as primary source of information.

Notice: trust Moodle over SISU

Please trust course Moodle as primary source of information.

We do not follow SISU weekly exercise (WE) signings, as there was problem that students were rejected because no room in volunteer weekly exercise groups, and many signees may not actually take this course now. So those signings may not be accurate, and limit actual attendees out.

At Moodle you sign to (volunteer) weekly exercises, but sign only if you are going to attend. Unfortunately every year we have had many tens of WE signees who never show up.

WE signing opens after first lectures (26.08. at 1605).

At Moodle you sign in for project assignment (= exercise groupwork).

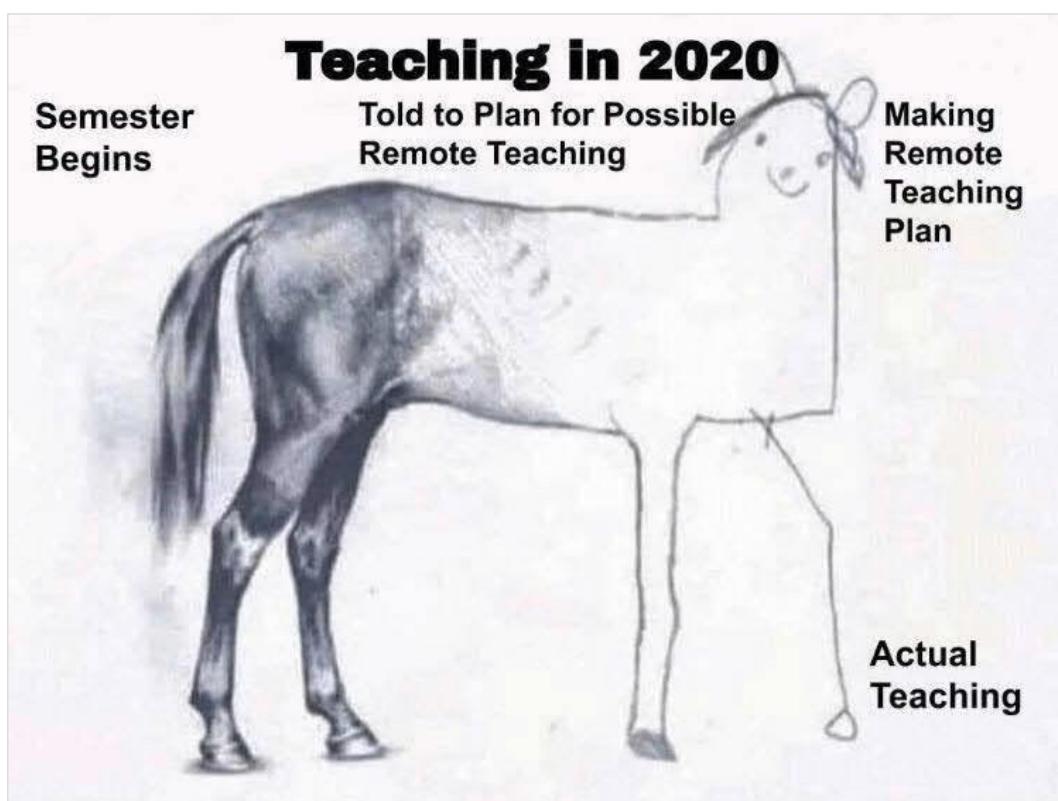
Feel free to search SISU about other COMP.SE courses of your interest, there is a lot of courses available in two (or three) campuses.

COMP.SE = software engineering

COMP.CS = computer science

At least if that (information technology or computer science) is your major (or long minor) study subject.

Hopefully that is not anymore the situation at TUNI in Autumn 2020.



Tampere is one of the leaders in ICT !

Daily Finland

[Print it](#)

Trial of coronavirus app Koronavilkku continues

11 Aug 2020, 11:56

 DF Report



Press Release Photo by National Institute for Health and Welfare (THL).

A mobile app designed to support the prevention of chains of coronavirus infection has been named Koronavilkku, said the National Institute for Health and Welfare (THL) in a press release.

The name was selected from among several proposals.

"We coined the name Koronavilkku together, reflecting the way the app itself was created in cooperation. The name was selected for its clarity and familiarity," said Ministry of Social Affairs and Health Communications Unit Director Vivikka Rieht.

The app will allow people to participate in preventing the spread of the virus and to protect their own and their loved ones' health. Through the app, people exposed to the virus can be reached and infection chains can be stopped faster.

Trial use of the app began on Tuesday, 4 August and will continue until the end of this week.

"Trial use ensures that all components of the app work together. The phones of test users in the Pirkanmaa Hospital District and the City of Helsinki have collected data on encounters. Test users have reported imaginary infections using the app, and the app has given other users notifications of potential exposure to the coronavirus," said THL Information Services Director Aleksi Yrttiaho, adding that the final element of the system, which is a user interface for professionals, will become available this week.

Information-dissemination events about the app were organised for social welfare and healthcare professionals jointly by the software company Solita, the Ministry of Social Affairs and Health, and SoteDigi.

"So far, less than 500 people have participated in them. Actual training for professionals will start next week," said Yrttiaho.

The general deployment of Koronavilkku is to take place in early September. At that point, anyone will be able to download this free app to their phone. Finnish and Swedish versions of the app will become

TUNI * COMP.SE.100-EN Introduction to Sw Eng

25.08.2020

13

Normal sw
business ?

Why ?

How to
avoid ?

 DATA CENTRE SOFTWARE SECURITY DEVOPS BUSINESS PERSONAL TECH SCIENCE EMERGENT TECH BOOTNOTES VENDOR VOICE  

[* EMERGENT TECH *]

Single-line software bug causes fledgling YAM cryptocurrency to implode just two days after launch

Day one: We're worth half a billion! Day Two: We're nearly worthless

Thu 13 Aug 2020 // 20:04 UTC 99  GOT TIPS?

Thomas Claburn in San Francisco   

 SHARE

A two-day-old decentralized cryptocurrency called YAM collapsed on Wednesday after its creators revealed that a software bug had effectively vetoed human governance.

"At approximately 6PM UTC, on Wednesday, August 12, we discovered a bug in the YAM rebasing contract that would mint far more YAM than intended to sell to the Uniswap YAM/CRV pool, sending a large amount of excess YAM to the protocol reserve," the YAM project explained in a post on Thursday.

"Given YAM's governance module, this bug would render it impossible to reach quorum, meaning no governance action would be possible and funds in the treasury would be locked."

The bug followed from this line of code...

```
totalSupply = initSupply.mul(yamsScalingFactor);
```

...which was supposed to be...

```
totalSupply = initSupply.mul(yamsScalingFactor).div(BASE);
```

[https://www.theregister.com/2020/08/13/yam_cryptocurrency_bug_governance/]

TUNI * COMP.SE.100-EN Introduction to Sw Eng

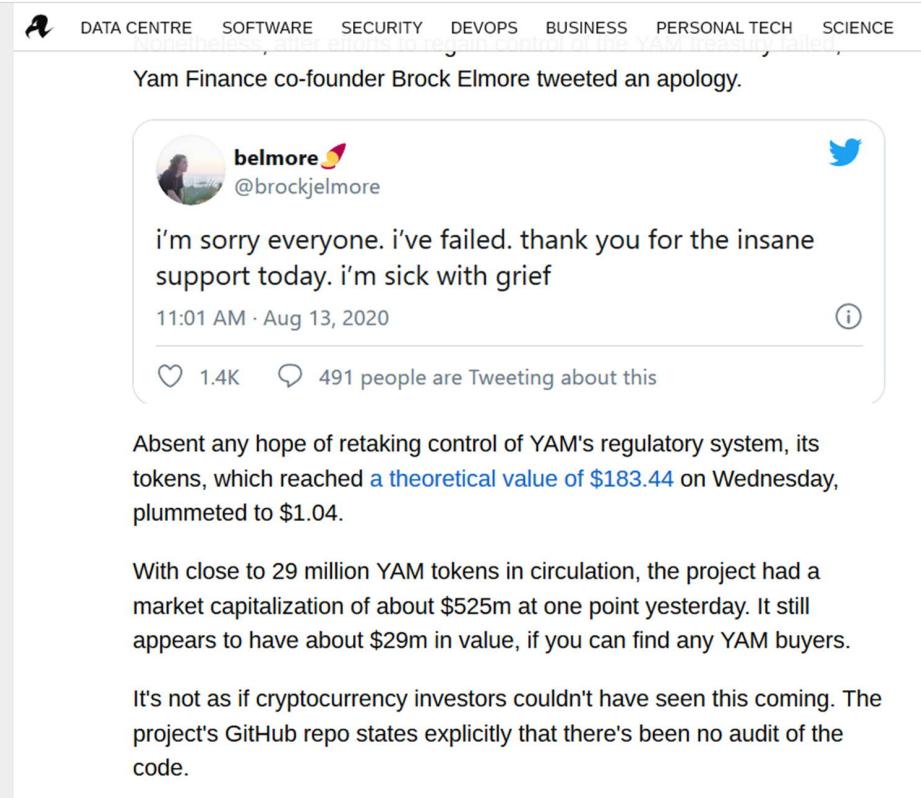
25.08.2020

14

Well...

there is never too
much testing...

but it is another
problem...



DATA CENTRE SOFTWARE SECURITY DEVOPS BUSINESS PERSONAL TECH SCIENCE E

Yam Finance co-founder Brock Elmore tweeted an apology.

 belmore @brockjelmore

i'm sorry everyone. i've failed. thank you for the insane support today. i'm sick with grief

11:01 AM · Aug 13, 2020

1.4K 491 people are Tweeting about this

Absent any hope of retaking control of YAM's regulatory system, its tokens, which reached [a theoretical value of \\$183.44](#) on Wednesday, plummeted to \$1.04.

With close to 29 million YAM tokens in circulation, the project had a market capitalization of about \$525m at one point yesterday. It still appears to have about \$29m in value, if you can find any YAM buyers.

It's not as if cryptocurrency investors couldn't have seen this coming. The project's GitHub repo states explicitly that there's been no audit of the code.

COMP.SE.100-EN Intro to Sw Eng

Course structure quickly in nutshell

- **lectures** (volunteer), remote at Zoom
 - pdf slides will be at Moodle (at lecture start time)
 - lecture recordings will be available at Panopto
 - **project assignment** (= exercise work), groups of 4 students
 - sign in groups at Moodle
 - **weekly exercises** (volunteer), Zoom and break room groups
 - sign in groups at Moodle
 - no homework, tasks done in small groups at WE
 - **exam**, three small EXAM exams.
-
- 2nd and 3rd examinations (2021) will be "one big" EXAM.

Course contents

1. Course basics, intro
2. Sw Eng in general, overview
3. Requirements
4. Different software systems
5. Basic UML Diagrams ("Class", Use Case, Navigation)
6. Life Cycle models
7. UML diagrams, in more detail
8. Quality and Testing
9. Project work
10. Project management
11. Open source, APIs, IPR
12. Embedded systems, IoT
13. Recap, summary

w36 lecture time
is explaining
project assignment

1. Course basics, intro

- course goals
- course personnel 2019
- course requirements
- grading
- lectures
- weekly exercises
- project assignment = exercise work
- hints for group work
- what to do next ?

"ItSE" course goals

This is the basic course to Software Engineering (Sw Eng), suitable also for minor study subject readers.

Learning outcomes of the course: After the course student understands how different types of software is developed in a professional way and what kind of knowledge and skills it requires. The ultimate goal is efficient participation in software projects in various stakeholder roles, for instance as an expert or customer. **We want you to understand the whole software development life cycle (SDLC) and its processes, which is much more than just coding in the dark basement.**

"From internet you can get information, but knowledge and wisdom you get from lectures."

Everybody should have an understanding about software development life cycle and processes. COMP.SE.100 gives you that view, as well as a more detailed touch to requirements, which are needed and understood by all stakeholders in every sw eng project. As in many projects the seeds of failure are planted in the beginning.

COMP.SE.100 EN (Autumn 2020) COMP.SE.100 FI (Spring 2021)

Two course implementations; English and Finnish.

Same contents.

Almost same material... still many English slides in the Finnish course. Not all slides are translated (Finnish news to English, English pictures/screenshots to Finnish).

Kaksi eri toteutusta, englanniksi ja suomeksi.

Sama sisältö.

Melkein sama materiaali... kaikkea en ole vielä ehtinyt käänämään sille toiselle kielelle. Eikä kaikkea voi käänää (esim. suomenkielisiä lehtileikkeitä englanniksi, tai englanninkielisiä tekstititetttyjä kuvia/kuvakaappauksia suomeksi).

History of the course

1990- 81210 "OTuPK"

... TIE-02300 ...TIE-02301 ... "JOTU", the basic sw eng course

2019- TIE-02306 Introduction to Software Engineering "ItSE"
Now the second English implementation, 2020.

At the Finnish JOTU courses there have been some 250..300 students annually; about 100 ICT students and 150 from other faculties. So later at your work many of you will very likely be involved in some ICT project as a customer, not as a software professional (in some vendor/developer company).

Spring 2020 (FI): about 350 students, more than 100 from City centre campus.

Course information

All course information (primary source) is in TUNI Moodle;

<https://moodle.tuni.fi/course/view.php?id=11284>

Padlet page (for all kind of questions):

<https://padlet.com/tensu/gzhwiol5clqkgjpk>

You may ask any time, in English or Finnish, I answer in English.

Vote deadline for project assignment

Vote at Moodle:

When (weekday and time) would be the deadline for project assignment (exercise work) deliveries ?

- Friday at 1600 o'clock
- Friday at 2355 o'clock
- Saturday at 2355 o'clock
- Sunday at 2355 o'clock.

REMEMBER: you can always return your work earlier, you need not to wait for the last minute. And sometimes, because of network delays, some last-minute returnings have missed the deadline. Oh boy.

Voting is open: 20.08. from 0800 to 31.08. to 2355 o'clock.

First SISU term... (2020)

Sorry about SISU, teachers may have not known how to use it correctly, when course data was entered in February.

Quite a many signees we have currently:

- at SISU: 161
- at Moodle: 186 students.

Last year we had about 50 (POP/ROCK) and 30 Moodle signees.

I only wonder if you all are REALLY going to take this (second ever) English implementation ?

ItSE course staff 2020

Course responsible person and lectures:

- Tero "Tensu" Ahtee, tero.ahtee@tuni.fi

Head/Lead assistant:

- Ulla-Talvikki Virta.

Weekly exercises:

- Sanna Hakojärvi
- Juanita Kananen
- Kevin Mäkelä.

Project assignment (exercise work):

- Lauri Hiltunen
- Aleksius Kurkela.

ItSE 2020 course staff



Course responsible,
tero.ahtee@tuni.fi

Tensu



Weekly exercise and
Project assignment master,
ulla-talvikki.virta@tuni.fi

Ulla



Sanna



Juanita



Kevin



Lauri



Aleksius

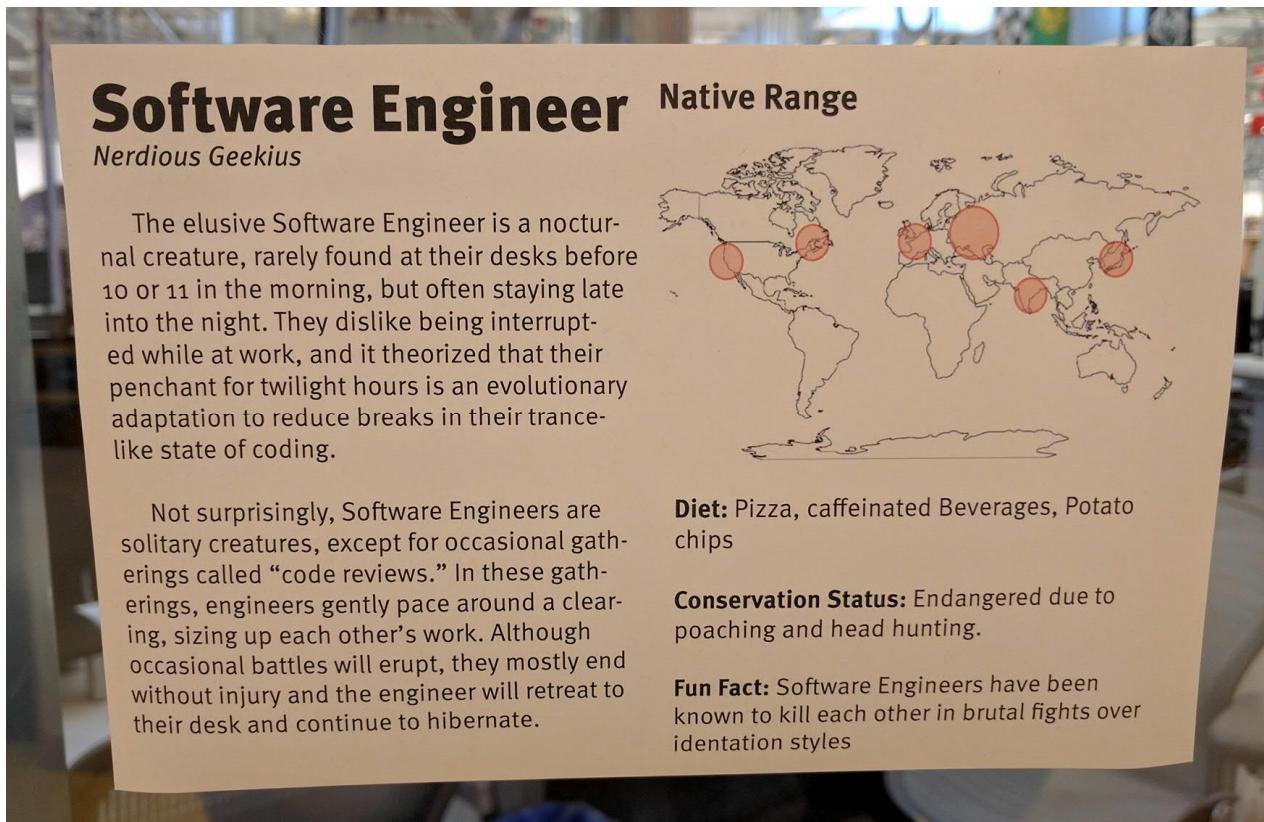
week	lectures	exam	weekly exercises	project assignment (exercise work)
35	L1: course basics		--- sign to WE groups ---	sign for project = grouping...
36	Project Assignment explained		WE1: intro to requirements	grouping, groups to Moodle
37	L2: Sw Eng in general		WE2: Trello and agile way	group's Trello board ready with product backlog
38	L3: requirements		WE3: feasibility study and stakeholder analysis	working...
39	L4: basic UML diagrams		WE4: requirements	working...
40	L5: more UML diagrams	EXAM-1	WE5: UML diagrams - Use case	working...
41	L6: different sw systems	EXAM-2	WE6: UML diagrams - concept/entity and navigation	deadline for 1st phase documentation and presentation
42	examination week		examination week	examination week
43	L7: life cycle models		groups' 1st presentations	groups' 1st phase presentations
44	L8: quality and testing	EXAM-2	WE7: development processes	feedback group-to-group at PRP, from 1st phase
45	L9: project work	EXAM-2	WE8: testing and error reporting	deadline for diagrams first versions
46	L10: project management		WE9: effort estimation	feedback to groups from diagrams (from assistants)
47	L11: open source, APIs, IPR		groups' 2nd presentations	working...
48	L12: embedded systems, IoT	EXAM-3	WE10: delivery contracts and terms of use	working...
49	L13: recap, summary	EXAM-3	---	final delivery of project documentation
50	examination week		examination week	feedback inside group, student-to-student at PRP
51	examination week		examination week	end of game / game over.
	Lectures: Wed at 1415-16.		Weekly exercises:	
			Mon 0815-10	AUTUMN 2020 (1-2. periods)
			Mon 1215-14	are remote/distant learning.
			Tue 0815-10	
			Tue 1415-16	
			Wed 0815-10.	

In general...

If you don't understand something (e.g. at Moodle page or project assignment guidelines), **ask** !

At corridors all kind of rumours and "war-stories" exist about university courses, heard from students who have heard those from some "reliable source". Oh boy. ; -)

It is much better to ask before acting or neglecting something. The same way is with software requirements.



25.08.2020

TUNI * COMP.SE.100-EN Introduction to Sw Eng

25.08.2020

29

In software engineering...

there is no magic

miracles do not happen

so YOU have to do the thinking and work YOURSELF.

So you want to be an engineer...

Find at YouTube "Dilbert knock the engineer"

...who also understands software processes (software economics and business you will learn at other units/faculties).

By the way, in software engineering there shall be nothing that you describe like "believe", "think", "hope that".
Everything must be "exact as engineering science", that is something you can measure.

(FI: ohjelmistotuotannossa ei ole uskon ja toivon asioita.)

Software engineers are needed in future !

≡ **POPULAR SCIENCE** [SUBSCRIBE](#)

CARS

SOFTWARE NOW TO BLAME FOR 15 PERCENT OF CAR RECALLS

YOU CAN'T JUST HOLD THE HOME AND LOCK BUTTONS TO SOLVE THIS ONE
 Bengt Halvorson / The Car Connection June 2, 2016

Self-driving Uber car kills Arizona pedestrian, police say

TRANSPORTATION \ UBER \ RIDE-SHARING

Uber's self-driving car showed no signs of slowing before fatal crash, police say

The vehicle was traveling at 40 mph

By Andrew J. Hawkins | @andyjayhawk | Mar 19, 2018, 7:09pm EDT

Tesla driver dies in first fatal crash while using autopilot mode

The autopilot sensors on the Model S failed to distinguish a white tractor-trailer crossing the highway against a bright sky

OPINION

Uber self-driving car accident: Who's to blame when there's no driver?

The Conversation By Raja Jurdak and Salil S. Kanhere

Posted Tue at 7:55am

TUNI * COMP.SE.100-EN Introduction to Sw Eng

25.08.2020 33

Software engineers are needed in future !

WIR ED

What Boeing's 737 MAX Has to Do With Cars: Software

AARON MARSHALL TRANSPORTATION 06.30.19 07:00 AM

WHAT BOEING'S 737 MAX HAS TO DO WITH CARS: SOFTWARE

THE WALL STREET JOURNAL

Home World U.S. Politics Economy Business Tech Markets Opinion Life & Arts Real Estate

BUSINESS

FAA Finds New Software Problem in Boeing's 737 MAX

Plane maker agrees to address the problem and believes it can be fixed with a software tweak

By Andrew Tangel and Andy Pasztor

Updated June 26, 2019 9:55 pm ET

Boeing Co. and federal regulators said they have identified a new software problem on the 737 MAX, further delaying the process of returning the troubled jet to service.

BUSINESS NEWS JUNE 27, 2019 / 5:30 PM / 2 MONTHS AGO

Boeing sees fix for latest 737 MAX software flaw in September

David Shepardson, Eric M. Johnson, Tracy Rucinski

5 MIN READ  

(Reuters) - Boeing Co. (BA.N) will take until at least September to fix a newly identified problem on its grounded 737 MAX, a company official told Reuters, meaning the workhorse jet's return to service will be delayed until October at the earliest, significantly longer than most airlines had expected.

Bloomberg

cnn politics 45 Congress Supreme Court Facts First 2020 Election

Technology

Boeing's 737 Max Software Outsourced to \$9-an-Hour Engineers

By Peter Robison

June 28, 2019, 11:46 PM GMT+3

New flaw discovered on Boeing 737 Max, sources say

By Shimon Prokupecz, Drew Griffin and Gregory Wallace, CNN

Updated 0719 GMT (1519 HKT) June 27, 2019

TUNI * COMP.SE.100-EN Introduction to Sw Eng

25.08.2020 34

Course structure and requirements, 1

- Lectures (volunteer)
 - Zoom
 - slides in Moodle (pdf)
 - lecture recordings via Panopto
- Weekly exercises (WE, volunteer)
 - Zoom, Break rooms for groupworking
 - supports lectures and exercise work
 - no homework, stress yourself only in WE times ;-)
 - presence logging by Jyke's QR-välppä
- Project assignment = exercise work (practical work)
 - made in groups of four (4) students
 - requirements specification documentation
 - UML diagrams: Concept (Entity), Navigation, Use Case
 - tools: [Trello](#), [Dia](#) ([/Enterprise Architect](#)), [PRP](#), ...
- Exam (electronic EXAM), 3 short exams during the course.

Course structure and requirements, 2

- Weekly exercises (WE, volunteer)
 - supports lectures and exercise work
 - no homework, stress yourself only in WE times (at Zoom)
 - most of work is made in small groups
 - presence logging by QR-code.

If you plan to attend weekly exercises, apply/sign for some exercise class. Signing at Moodle opens after first lecture (26.08.2020 at 1605 o'clock).

[WE signing at Moodle ends on Sun 30.08. at 2359 o'clock](#). Depending of the amount of students at this course and sign-ups for WE groups/classes, we may consider the amount of WE groups (currently five).

[Responsible master assistant: Ulla.](#)

Course structure and requirements, 3

- Project assignment = exercise work (practical work)
 - made in groups of four (4) students
 - requirements documentation (not exactly a full specification)
 - made in two phases
 - first (1.) presentation and final (2.) presentation
 - UML diagrams: Use Case, Concept (Entity), Navigation
 - document delivery in pdf format at TUNI Moodle
 - presentation slides' delivery in pdf-format at PRP (via Moodle).
- tools:
 - [Trello](#) "project management"
 - [Dia](#) (/Enterprise Architect) for UML diagramming
 - [PRP \(Peer Review Program\)](#) peer feedback, self assessment.
- [Responsible master assistant: Ulla.](#)

Project Assignment = Exercise Work

Don't be scared about the amount of documentation. Every graduated MSc has to write some reports and documents, sooner or later. And BSc thesis before that.

Well, everything you do for the first time,
seems to require a lot of work and be difficult.
But the second time later on will be much easier. ;-)
So you have better to start early.

Course structure and requirements, 4

- Exam (electronic EXAM, or Forms or Moodle in emergency)
 - three (3) small exams during the course, no "one big bang" at the end.
 - one exam will be about diagrams (Dia tool should be at EXAM classes).
1. **Sw Eng basics and requirements** (weeks 40-41)
 2. Diagrams (Dia tool at EXAM classes) (weeks 44-45)
 3. Other relevant Sw Eng matters (weeks 48-49).

2nd and 3rd examination possibilities (to repeat exam) would be one large exam covering the whole course, at some times during Spring 2021.

Grading

Student collects points from

- project assignment
- exam(s)
- volunteer weekly exercises.

From exam a certain minimum amount of point is required to pass.

From project a certain minimum amount of point is required to pass.

Total grade (fail, 1-5) comes by amount of points gathered along the course.

Points for grading the ItSE course 2020

- project assignment; max. 13 (PRP included), minimum requirement at least 5 points
- exams; max. 18 (6+6+6), minimum at least 2 points from every exam
- weekly exercises (volunteer); max. 5 points in total
- totalling 31/36 points at maximum.

Grading:

- 1: 15-
- 2: 20-
- 3: 24-
- 4: 27-
- 5: 30-

Points and scaling at the moment, 13 Aug 2020.

Late delivery (missing deadline) of work reduces points.

Kahoot! Quiz. Or Survey.

Are you going to attend Weekly Exercises ?

How many years you have studied at university ? 0 this is first, 1, 2, 3or+

Is Software Engineering: Science, Art, Folklore ?

Have you used tools; Trello, Dia, PRP earlier ?

These are by purpose...

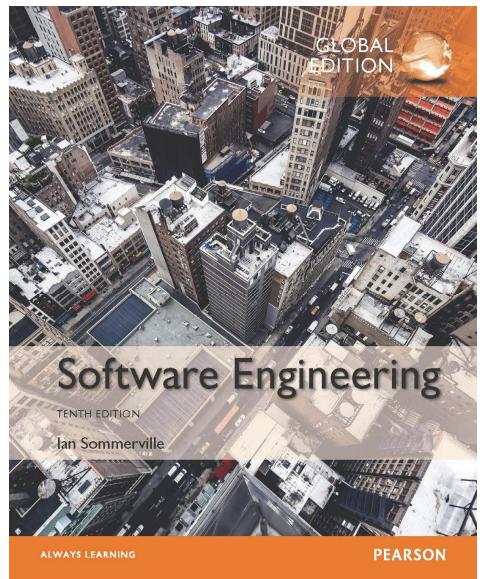
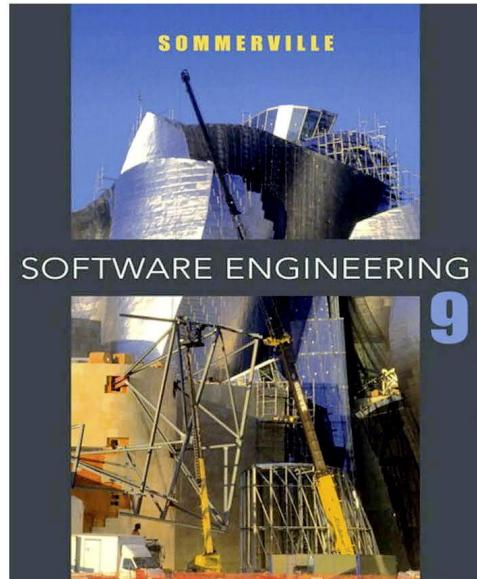
- some matters are being presented a few times, because those are essential, so take those "duplicate slides" as refreshment for memory
- a lot of material i.e. slides on the lectures, I try to pick essential (specially for minor readers) first and then "nice to know" (for major readers)
- A lot of slides; sometimes students have complained that "I already know all these as I have been working six months in a software company". Well, I doubt that knowledge.
- I know that pedagogically it would be good to show only one "truth" to students, but in software engineering there is very seldom just one "right" way to do things. That's why in many cases there are several slides about the same subject.
- And yes, I may use **two** or **three** colours in texts at a same slide, by purpose to make things **visible**.

Sw Eng books

There are tens if not hundreds of "Software Engineering" textbooks.

Some can be found from TUNI electronic library as eBooks.

Ian Sommerville's book is the most popular one.



Famous books in sw eng



Other material

See also "Additional material" section at the end of course Moodle page. It includes a lot of miscellaneous links.

Note also TUNI electronic library

- eBooks (some Sw Eng books available)
- SFS Online (standards)
- IEEEXplore (IEEE publications).

Do not forget all kind of sw eng comics/cartoons, e.g.

- Dilbert
- geek-and-poke
- Glasbergen
- PhD comics.

www.scrumguides.org

The Scrum Guide™

The Definitive Guide to Scrum:
The Rules of the Game

November 2017



Jeff Sutherland

Ken Schwaber

Developed and sustained by Scrum creators: Ken Schwaber and Jeff Sutherland

"If you do agile methods by the book, you are not agile."

Scrum-opas™

Scrumin määritelmä ja pelisäännöt

marraskuu 2017



Jeff Sutherland

Ken Schwaber

Kirjoittajat ovat Scrumin kehittäjät Ken Schwaber ja Jeff Sutherland

SUOMI / FINNISH

Software Engineering -- ohjelmistotuotanto

Software (FI: ohjelmisto)

1. computer programs, procedures and possibly associated documentation and data pertaining to the operation of a computer system.
2. all or part of the programs, procedures, rules, and associated documentation of an information processing system.
3. program or set of programs used to run a computer.

Software Engineering (FI: ohjelmistotuotanto)

1. systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software.
2. application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

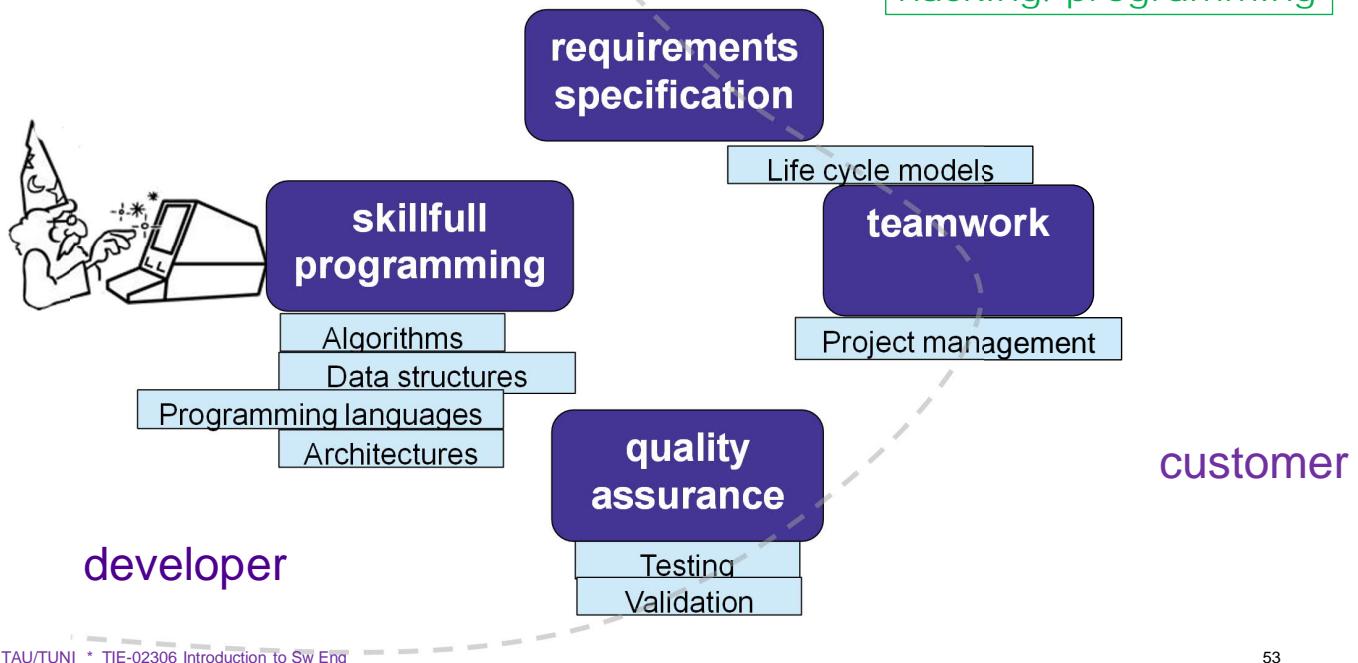
[IEEE 24765-2017]

Software engineering =
Requirements specefication
Programming
Quality assurance
Documentation
Project management
Deployment
Sales (and after sales)
Maintenance.

Sw Eng is NOT just
hacking/programming

Requirements specification =
Finding out needed/required features, and defining those.

What is Software Engineering ?



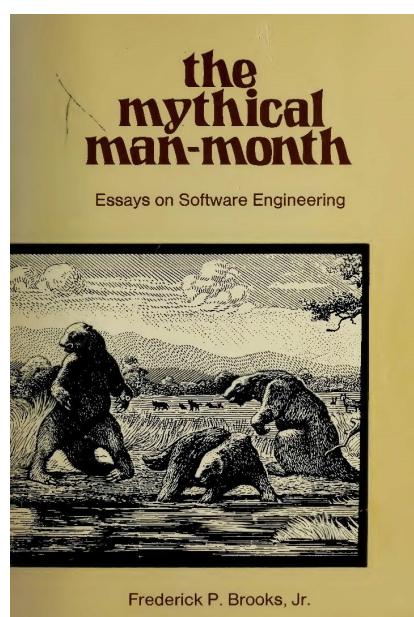
Why making software is so hard/difficult ?

Software is abstract (compare to building a house)

- developers and users have different views and opinions
- estimating effort is difficult.

Software is dynamic

- may be changed "easily"
- others suppose sw is "flexible"



Software development is not scalable

- adding more workers do not speed up much
- more workers = more communication
- small team tactics do not necessarily work with large sw projects.

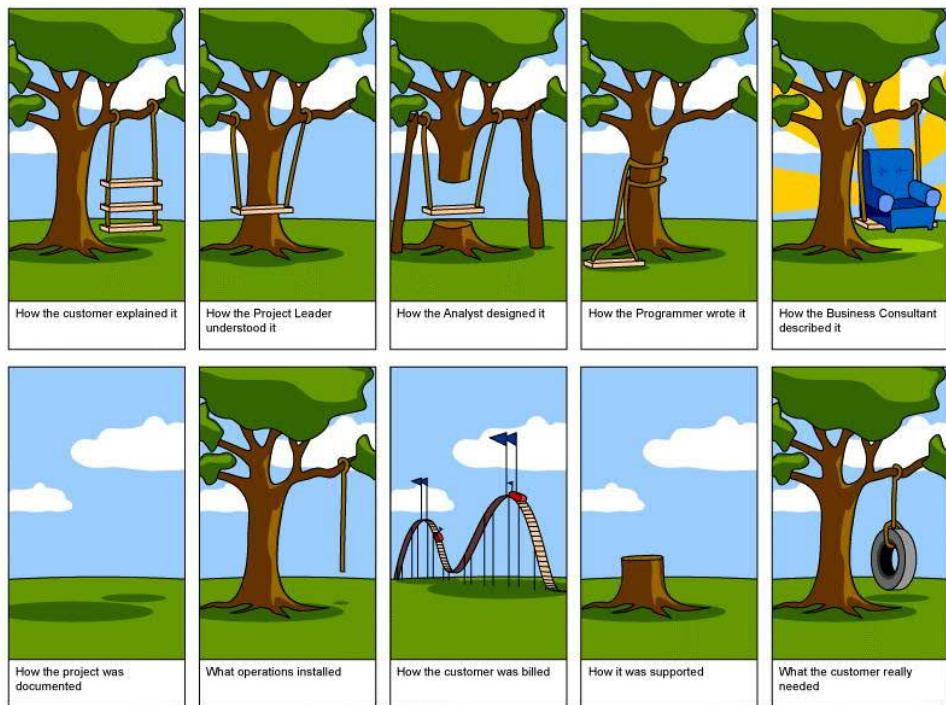
Main point at ItSE: requirements

Why so many software projects fail ?

In many cases it is about requirements and the process.

And in many such cases, it is also the customer's fault, too. Not committed.

An active and knowledgeable customer is needed for success !



developer vs. customer about the product

Well, good requirements are for example

- complete
- accurate
- unambiguous
- clear
- easy to understand
- short
- and much more...

That is easier to say than do.



Hint: don't argue, negotiate.

Some folklore

"Old versions of software are faster and more reliable than new ones."

Sounds familiar ?

Why is it so ?

Perhaps some 2..3 minutes Break rooms here !

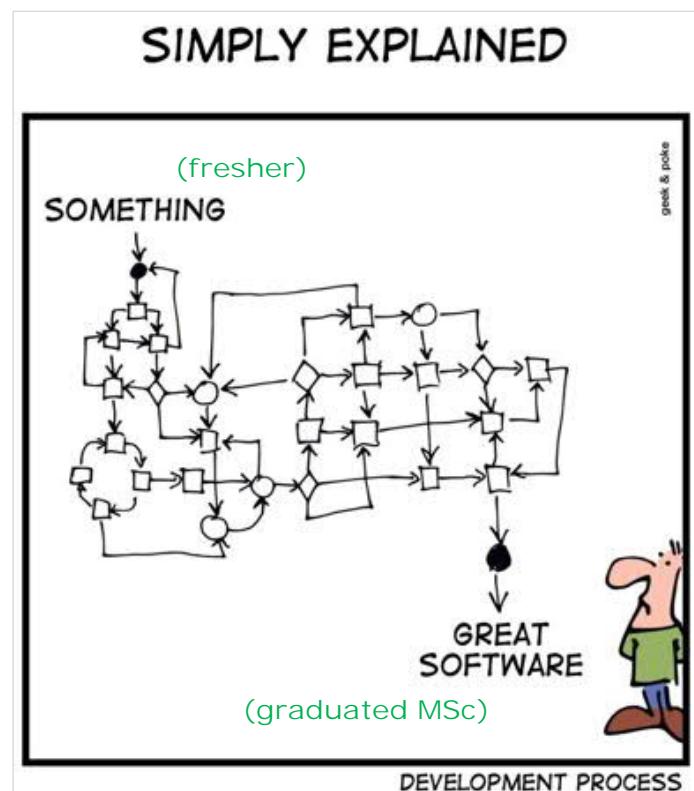
A good UI "rescues" user experience of a poor software.

A poor UI "destroys" user experience of a good software.

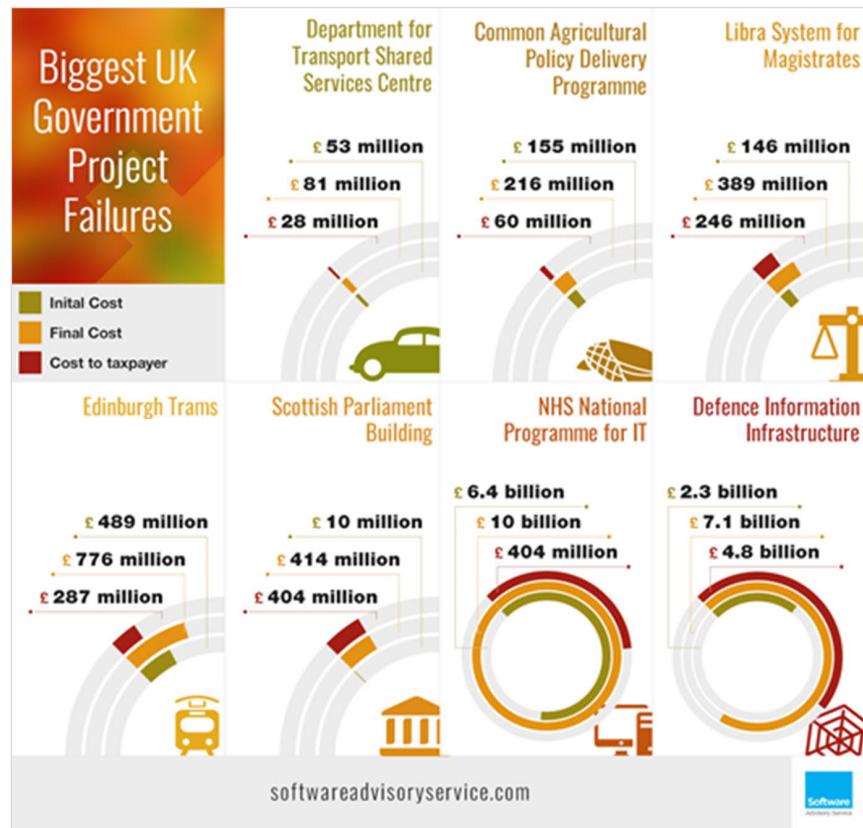
UI = user interface.

Well, of course you need a bit of good luck in software engineering projects... but the technical basics you need to know anyway.

Quite well-known method "Desperate hacking", in Finnish "Läpisekoilun periaate".



Something you would not like to read from newspapers... and definitely not for those projects YOU work for.



TUNI * COMP.SE.100-EN Introduction to Sw Eng

25.08.2020

59

SLC = software life cycle

software life cycle. The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The software life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and, sometimes, retirement phase. Note: These phases may overlap or be performed iteratively. Contrast with: **software development cycle**.

Now you think has anything changed from 1990s ?

Well, in every Sw Eng project there is some

- preliminary analysis
 - requirements specification } PLAN
 - design
 - implementation
 - testing
 - documentation.
- DO
- CHECK/TEST

But methods and processes, way of working, have changed.

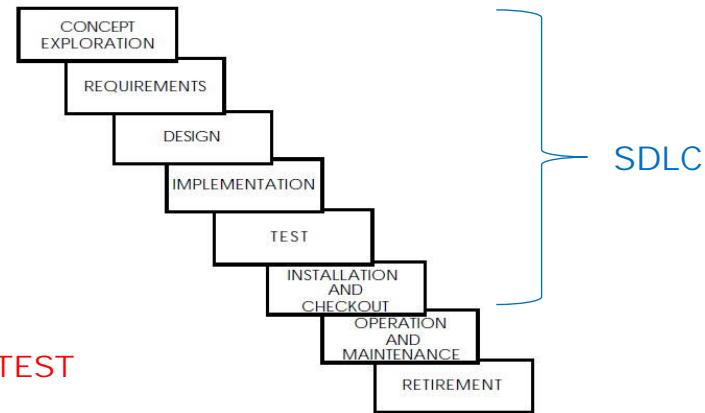


Fig 15
Sample Software Life Cycle

[IEEE Standard Glossary of Software Engineering Terminology, Std 610.12-1990(R2002)]

SLC = software life cycle

software life cycle. The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The software life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and, sometimes, retirement phase. *Note:* These phases may overlap or be performed iteratively. *Contrast with:* **software development cycle**.

COMP.SE.100
is about that
small part

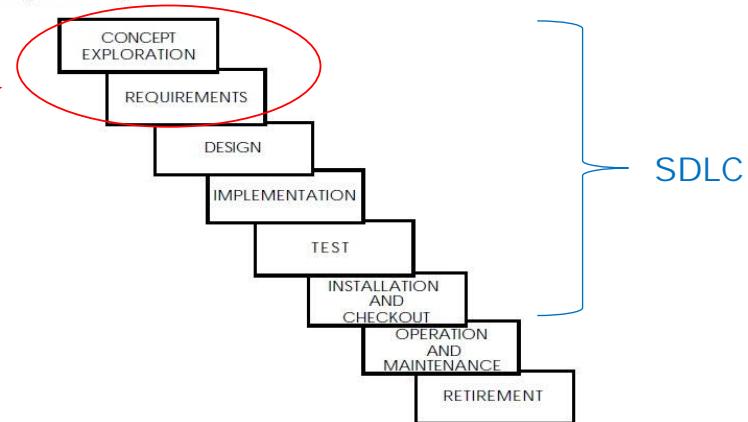


Fig 15
Sample Software Life Cycle

[IEEE Standard Glossary of Software Engineering Terminology, Std 610.12-1990(R2002)]

25.08.2020

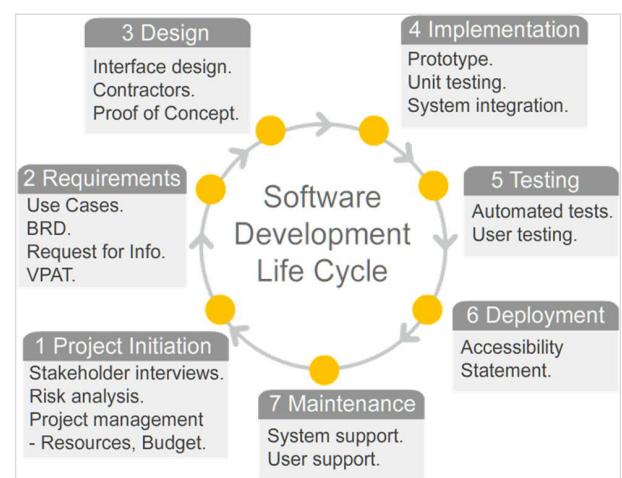
TUNI * COMP.SE.100-EN Introduction to Sw Eng

61

“Waterfall” shaped as a SDLC circle

Well, in every Sw Eng project there is some

- preliminary analysis
- requirements specification
- design
- implementation
- testing
- documentation.



In every life cycle model there are the same basic items in some role.

BRD= business requirements document
VPAT= voluntary product accessibility template.

25.08.2020

TUNI * COMP.SE.100-EN Introduction to Sw Eng

62

Software life cycle

ISO/IEC/IEEE 24765:2017(E)
Systems and software engineering — Vocabulary
Second edition, 2017

3.3803

software development cycle

SDC = SDLC

1. period of time that begins with the decision to develop a software product and ends when the software is delivered

cf. software life cycle

3.3823

software life cycle (SLC)

SLC

1. project-specific sequence of activities that is created by mapping the activities of a standard onto a selected software life cycle model (SLCM) [IEEE 730-2014 IEEE Standard for Software Quality Assurance Processes, 3.2]

2. software system or software product cycle initiated by a user need or a perceived customer need and terminated by discontinued use of the product or when the software is no longer available for use.

A good engineer is exact, but nothing beats common sense.

Misunderstandings are the most common problems in software projects (not the technical problems).



Finnish study among companies (2017)

What kind of skills are needed/required/wanted from new workers

1. Attitude, will to learn new, not giving up
2. Problem solving skills
3. Teamworking skills, communication skills.

In sw eng projects, TECHNICAL matters seldom break... it is the other matters that causes problems.

Also "soft skills" are needed in ICT projects !

A project with unclear requirements is like ...



A project with unclear requirements is like driving with poor visibility of your destination.

[Galileosystems]

Groupwork - is essential in sw eng

There will be a separate pdf file at Moodle about that.

Hints for group work

- contact other groupmembers quickly, get together
- find out information about the documentation and other tools; agree what to use, how those function, help-files and/or manuals
- **agree goals**; e.g. grade 1 or 5, or weekly hours effort
- work division; **Scrum Master (now also Project Manager)**
- a fixed (same weekday and time) **weekly meeting** is highly recommended
- perhaps a common calendar, for project work and times "not available" (e.g. exams, travelling, hobbies)
- **communication tools within the group** (e.g. slack, telegram, whatsapp, SMS, e-mail,...)
- perhaps project Start **Sauna Party** ("kick-off") ?

Causes of conflicts [Borg et al., 2011]

- ambition differences
- cultural differences
- bad communication
- strong wills
- unclear goals
- different prior knowledge (studies)
- aversion/dislike towards methods/tools.

Remember at group work...

If group-members have very different needs, it may lead to problems.



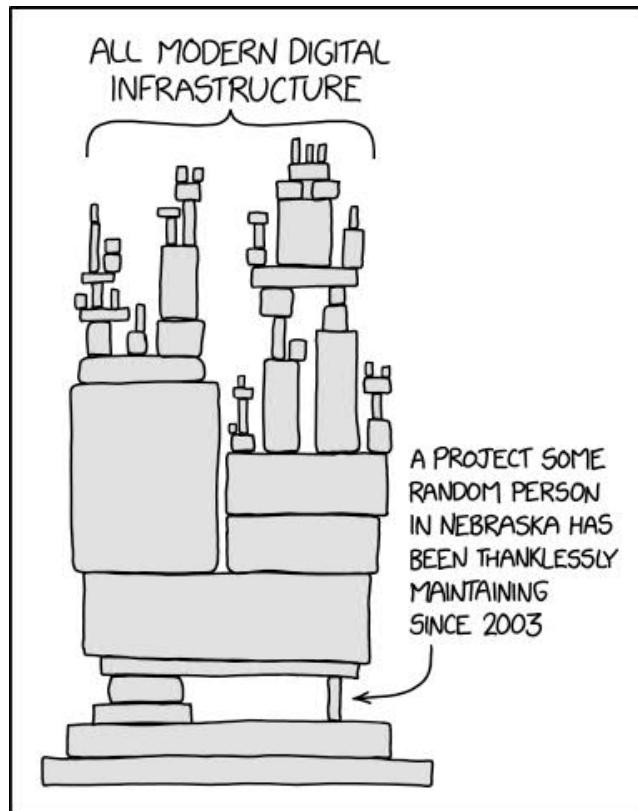
- state group rules early & clearly
- encourage to discuss problems
- well-defined roles
- start work with small tasks.

"World is full of friends you have not yet met."

You may well **co-operate** with fellow students,
but **NOT COPY** other's work.

What "time bombs"
may there be in old
(legacy) information
systems... some
unknown dependencies
perhaps ?

If you update one part,
does it break
something else ?



Agile way, Scrum

The Agile idea is (see Agile Manifesto)

- ask customer what (s)he wants = customer collaboration
- make the application in iterations ("small steps")
- after every iteration, show customer a demo, and listen comments
- change requirements if necessary = respond to change
- documentation is kept at minimum (but some have to be done).

Note that it is supposed that customer is active, too.

Agile way, Scrum

The Scrum idea is

- put all work items to Product Backlog (PB)
- take only few items to Sprint Backlog (SB)
- do one task at a time from SB
- after every Sprint you can show a demo to customer
- take customer feedback, modify PB if necessary.

Basicly, Scrum was mentioned to experienced teams, who have already been working together on some projects, so that they know each other's strengths.

Agile way, Scrum

Agile, good and yes, but do not run into hasty conclusions.

You still have to think a while before acting.

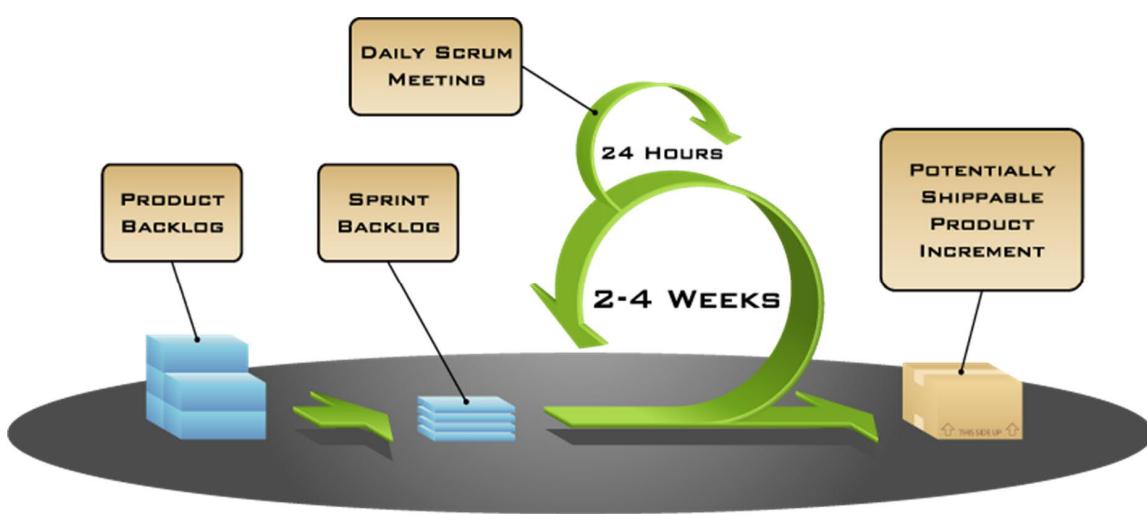
Check things first.

Agile or Scrum does not fit to all sw projects.

Occasionally old inventions are found and given a new "hype" name.

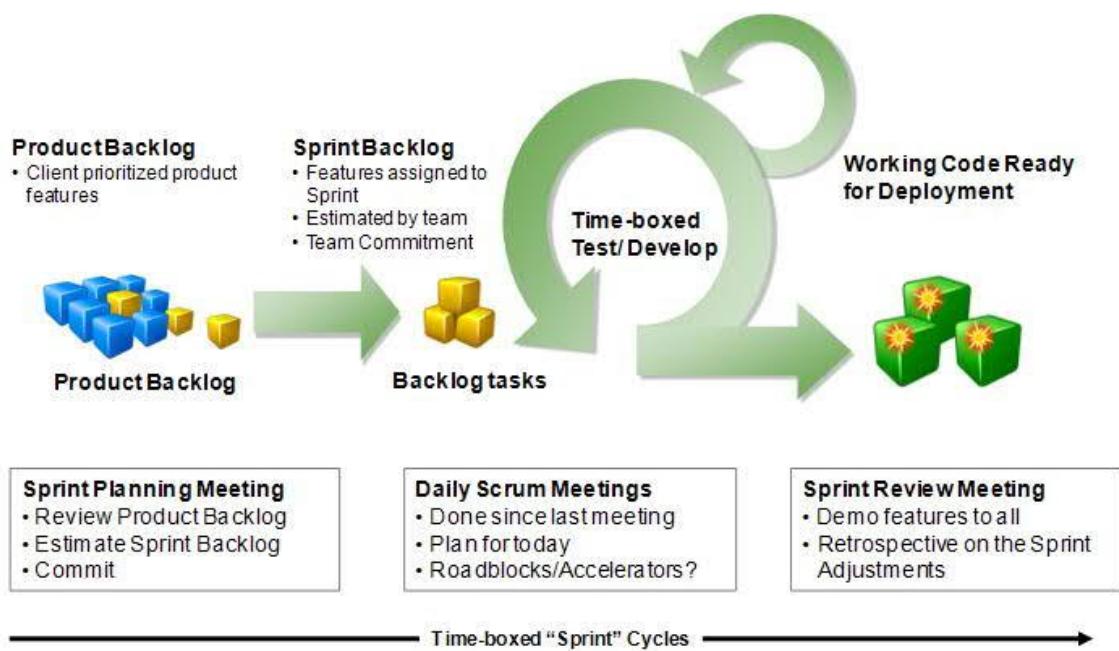


Putting it all together, iterative way



[www.mountaingoatsoftware.com/scrum]

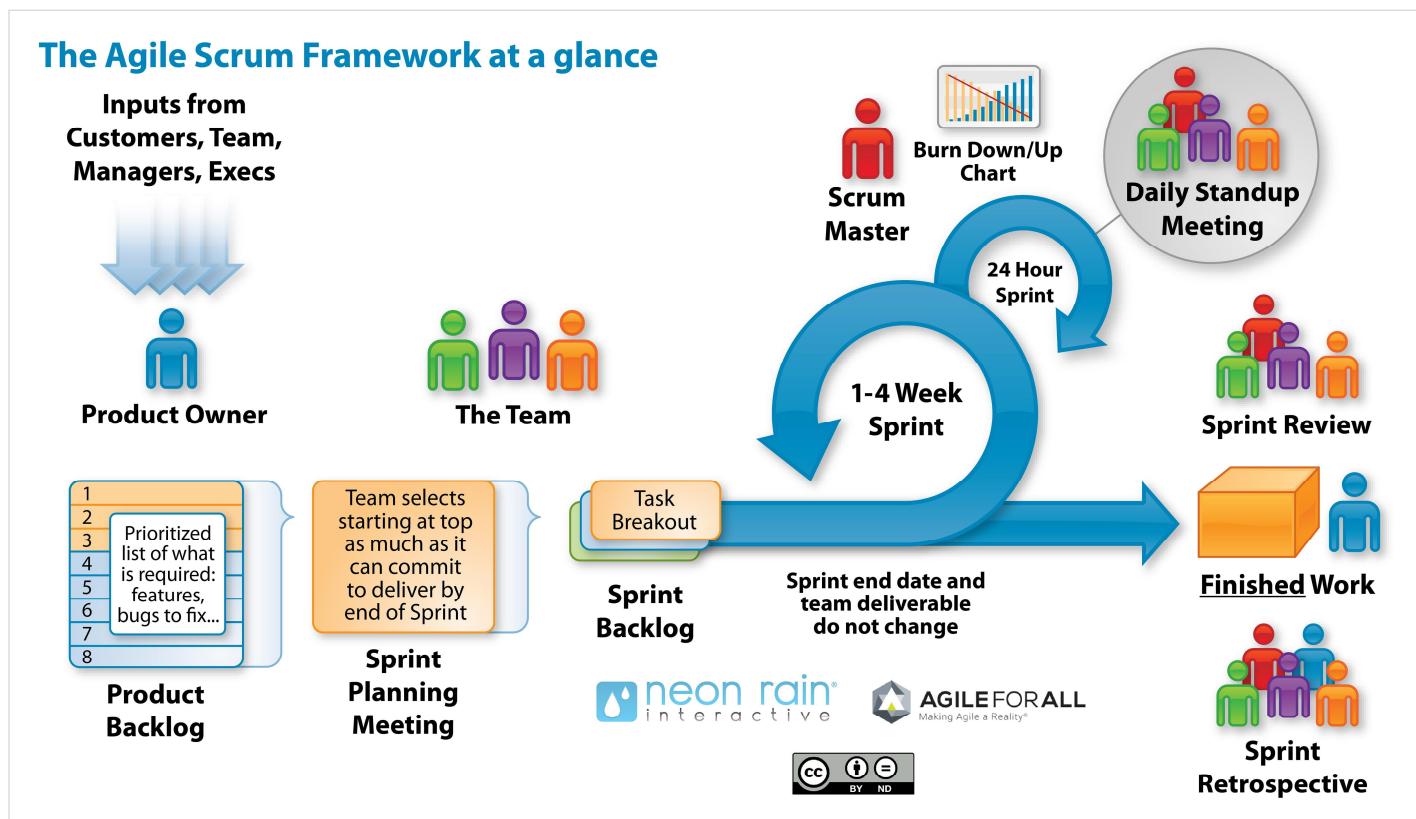
Scrum basics



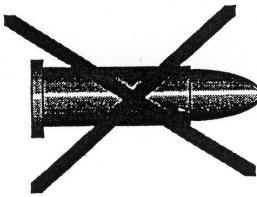
25.08.2020

TUNI * COMP.SE.100-EN Introduction to Sw Eng

77



There is no silver bullet!



No tool, no method will save you from having to think!



Problems in Sw Eng projects (study 2013)

Customer point of view:

- Schedule overrun
- Budget overrun
- Different thoughts about project goals
- Lack of quality

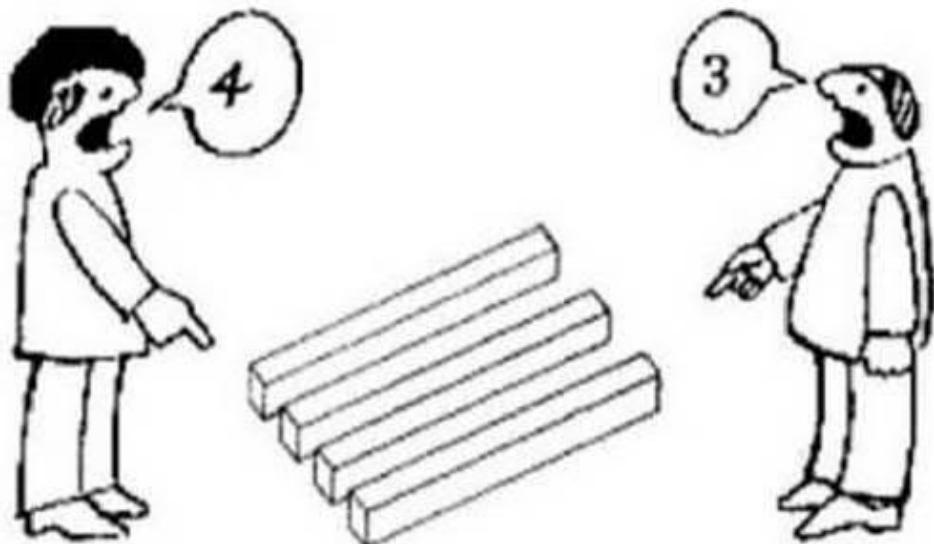
Vendor (developer company / software company) point of view:

- Lack of communication
- Different thoughts about project goals
- Schedule overrun
- Personnel changes
- Lack of quality.

Who is right ?

Developer

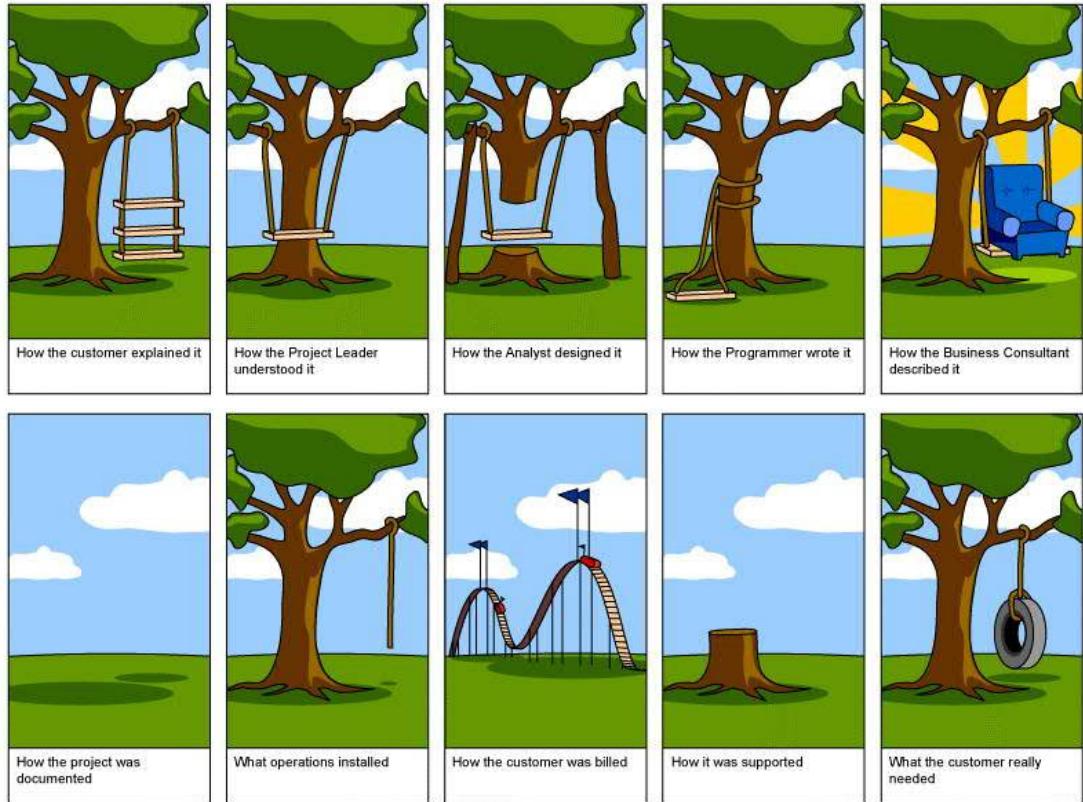
Tester or Customer



Famous phrase anecdote

“To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem.”

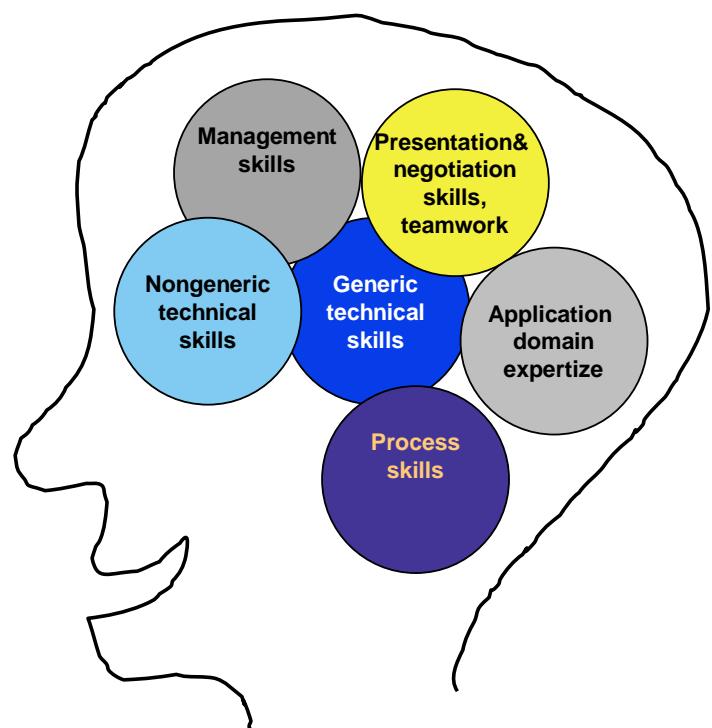
-- E. Dijkstra, 1972 Turing Award Lecture

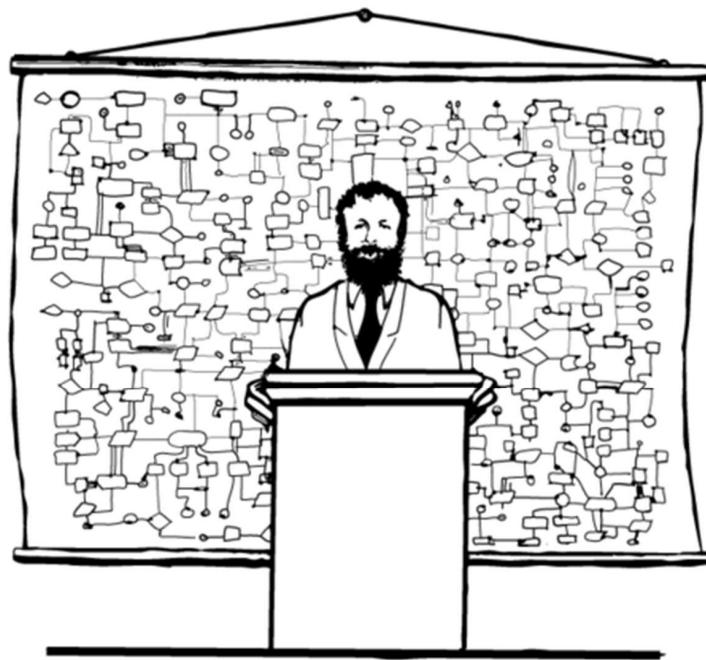


Sw Eng skills

Many kind of skills are needed for successfull software engineering...

much more than just programming or just technical skills.

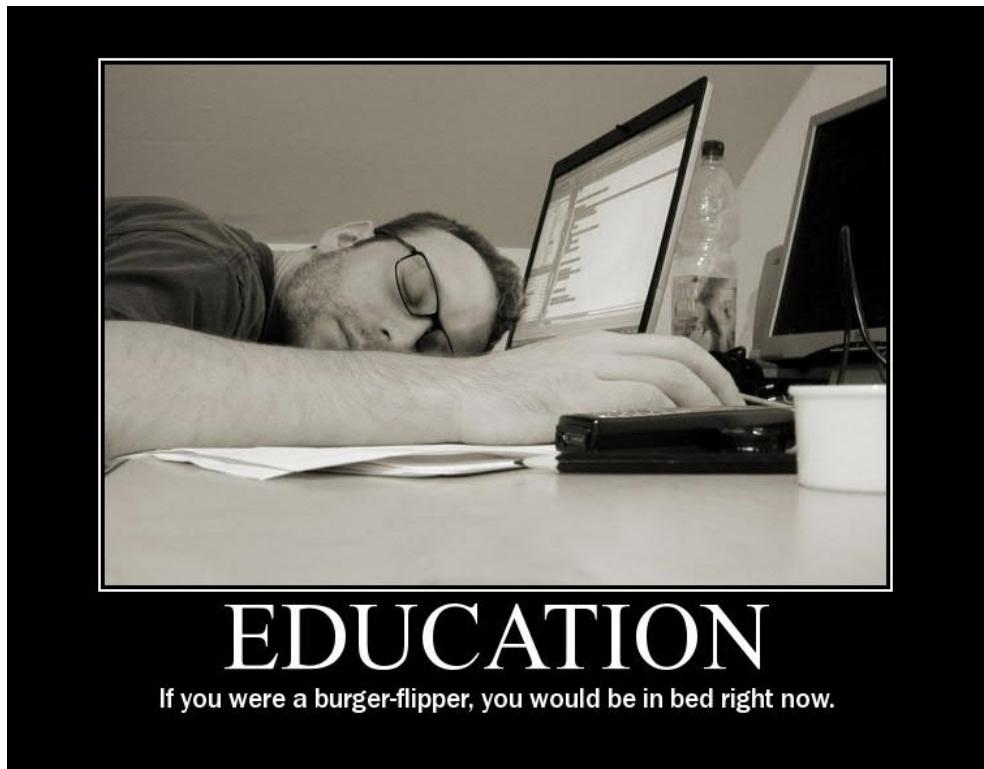




“Now that you have an overview of the system,
we’re ready for a little more detail”

What to do next...?

- find a project assignment group / search more students to fulfil your group
- you may sign in some weekly exercise group (unfortunately there will not be room in five groups for all 160 students), signing opens today at Moodle at 1605 o'clock
- we watch during this week 35 how full the WE groups will be



Summary / highlights

- software engineering is not magic, but stay alert
- misunderstandings are the most common reason for all kind of problems
- do not think you know what the other party/stakeholder thinks, ask
- when you get a project (idea), check first already existing similar solutions
- software is abstract, not easy to see from outside if it is working well
- there is never too much testing, but you can not test everything
- "fail fast, kill your darlings", i.e. make prototype quickly and change it
- never forget the actual end users
- you may use checklists/standards/best practices, many are available
- visit customer's/developer's company, learn the business (way of work)
- you may ask help from senior colleagues/workers, do not bang your head to wall
- how to eat an elephant... in small pieces, i.e. iterative work
- if you are required to handle HETU (social security ID), you are in trouble.

Remember also UN recommendations at all work



SUSTAINABLE DEVELOPMENT GOALS

